

## LA-UR-20-21551

Approved for public release; distribution is unlimited.

Title:	Scripting ToolKit (STK) 2.0 for Granta Database. Model Calibration Examples
Author(s):	Benedikt, Bartlomiej Jozef
Intended for:	Share it with Granta MI (company that provides Granta database to LANL)
Issued:	2020-02-18

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.



# **Scripting ToolKit (STK) 2.0 for Granta Database Model Calibration Examples**

2-12-2020

UNCLASSIFIED

# Motivation

Multiple stress cushion (SX-358) uniaxial compression tests are performed.

There are 640 ( $= 4 \text{ cycles} \times 160 \text{ tests}$ ) compression test curves coming to Granta.

Find CHIP-Foam material model parameters.

Efficient method is needed to process test data to calibrate custom material models.

UNCLASSIFIED

# Agenda

1. What is Python STK 2.0?
2. How can to access and use it?
3. Example 1: How to export data from Granta?
4. Example2: Model Calibration
  - CHIP-Foam Model
  - Updates to Driver
  - Calibration Example
5. Conclusions and Current Projects that Use STK 2.0

UNCLASSIFIED

# What Is Python STK 2.0?

**MI** Scripting Toolkit 2.0 for Python

Search: -Search-

- Welcome
  - System requirements
  - Upgrading from earlier versions
- Streamlined API Example Notebooks
  - 0 Get Started
  - 1 Get Attributes
  - 2 Search MI
  - 3 Fetch Attribute Data in Bulk
  - 4 Edit Data
  - 5 Create Records
  - 6 Import Functional Data
  - 7 Create Functional Data
  - 8 Edit Tabular Data
  - 9 Edit Pseudo-attributes
  - 10 Add Files and Pictures
  - 11 Create and Edit Tabular Data
  - 12 Combine APIs for FEA Export
  - 13 Link Records
- Streamlined API Reference
- Foundation API Documentation
- Access the log files
- Copyright and trademarks

## Welcome to the GRANTA MI:Scripting Toolkit 2.0 Help

The MI:Scripting Toolkit for Python provides an interface for directly interacting with the MI:Service Layer. Materials data in your GRANTA MI installation can be analyzed or transformed using Python, and the results imported back into MI with full traceability, or extracted based on known record lists or search criteria.

The Streamlined API introduced in Version 2.0 for Python builds on the existing Foundation API, providing wrappers for its request-and-response based classes and resulting in a more intuitive user experience. [Example Notebooks](#) are provided to help you get started, and all previous modules are still available and can be used alongside the new. For more specific assistance, the documentation is searchable and arranged into groups of classes and functions associated with MI [Sessions](#), [Databases & Tables](#), [Records](#) and [Attributes](#).

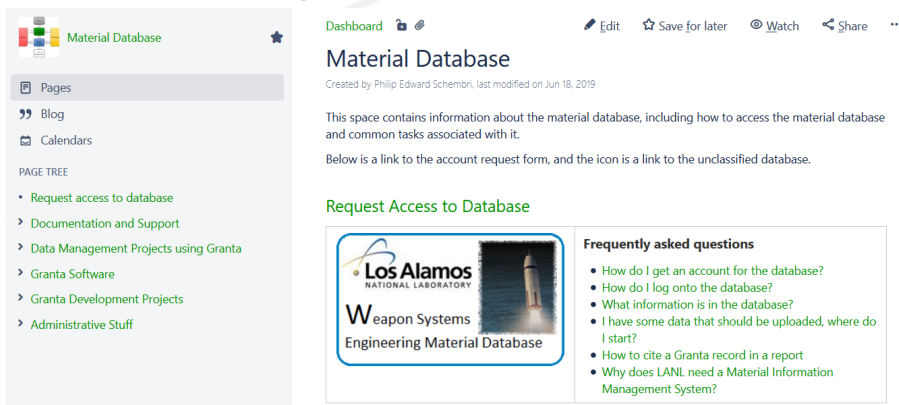
Minimal changes to your existing scripts are required, since the Foundation API classes are still included in the MI:Scripting Toolkit. More information can be found on the [Upgrading from earlier versions](#) page.

Python STK 2.0 is an interface for accessing Granta records from Python script.

It is similar to what Prabhu did a few years ago (Granta Query Daemon, GQD), however it has more features and it is supported by Granta MI.

UNCLASSIFIED

# What Is Python STK 2.0?



Material Database

Created by Philip Edward Schembri, last modified on Jun 18, 2019

This space contains information about the material database, including how to access the material database and common tasks associated with it.

Below is a link to the account request form, and the icon is a link to the unclassified database.

**Request Access to Database**

**Frequently asked questions**

- How do I get an account for the database?
- How do I log onto the database?
- What information is in the database?
- I have some data that should be uploaded, where do I start?
- How to cite a Granta record in a report
- Why does LANL need a Material Information Management System?

Several examples of how to perform basic tasks with STK 2.0 can be found on the Granta Confluence page.

## Where STK 2.0 Is Used:

- CHIP-Foam project (calibration of material model)
- TIMS: provides interface between TIMS database and Abaqus models
- DC-745: used to calibrate hyperelastic models (e.g. polynomial models)

## Granta Python STK 2.0 Example Scripts

File	Description	Author
STKv2_Test_WildcardAttrib.py (Windows/Euphrates-vm) STKInx_Test_WildAttrib.py (Linux)	Return all records in database that match a wildcard in any attribute. The initial script search criterion uses the attribute "Material Name" to search for records.	Clay
STKv2_Test_SpecificHeat.py (Windows/Euphrates-vm) STKInx_Test_Cv.py (Linux)	Get specific heat from a record in Material Universe	Clay
STKv2_Test_YoungsModulus.py (Windows/Euphrates-vm) STKInx_Test_Modulus.py (Linux)	Get Young's Modulus from a Common Material Property record in the LANL database	Clay
STKv2_pbx9502_EngSE_plot.py (Windows/Euphrates-vm) STKInx_Test_EngSEplt.py (Linux)	Get stress-strain curve from a PBX 9502 uniaxial test record in the LANL database	Clay
STKv2_Test_FileDownload.py (Windows/Euphrates-vm) STKInx_Test_FileDownload.py (Linux)	Download a file that is part of a record in granta	Clay
STKv2_Test_Images.py (Windows/Euphrates-vm) STKInx_Test_Images.py (Linux)	Download a picture that is part of a record in granta	Clay

UNCLASSIFIED

# How Can I Access/Use STK 2.0?

```
# Import STK 2.0 module:
import GRANTA_MIScriptingToolkit as GrantaToolkit
from GRANTA_MIScriptingToolkit import granta as mpy
```

Import STK 2.0

```
#### Input your username ( z# + "@lanl.gov")
user_name = "193644@lanl.gov"
#### Input your password (crypto-card)
cred = getpass.getpass("Password: ")
#
#
# Define INPUT for connecting to Granta, setting the units, etc:
# -----
#
#### Connect to Granta MI:
mi = mpy.mi.Session("http://grantami/mi_servicelayer", user_name, cred)
#### Connect to Database:
my_db = mi.get_db("LANL Weapon Systems Engineering Materials Database")
##### Set Unit System for Database:
my_db.set_unit_system('MPa-mm-s-K', absolute_temperatures=True)
#### Find Table in Database:
my_tbl = my_db.get_table("Testing - Uniaxial")
```

Open database and  
open the Table you  
want to access

STK 2.0 module is saved in beta version of Toolbox.

Beta toolbox can be accessed by *sv3b*.

STK 2.0 has also been installed on Windows *vm-euthrates*

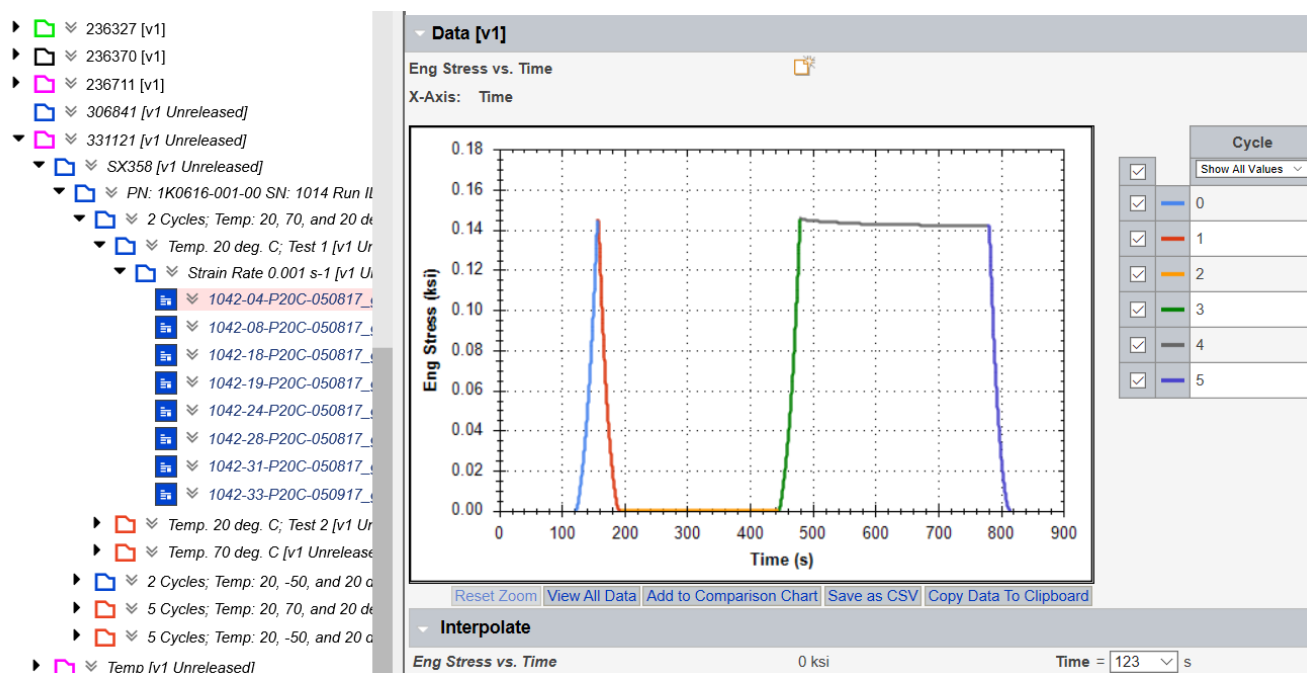
UNCLASSIFIED



# Example 1: How to Export Data from Granta?

Goal:

Need to export to Python stress vs. strain data from selected records.



UNCLASSIFIED

# Example 1: How to Export Data from Granta?

Specify what records to search, what functional data to export, units, etc.

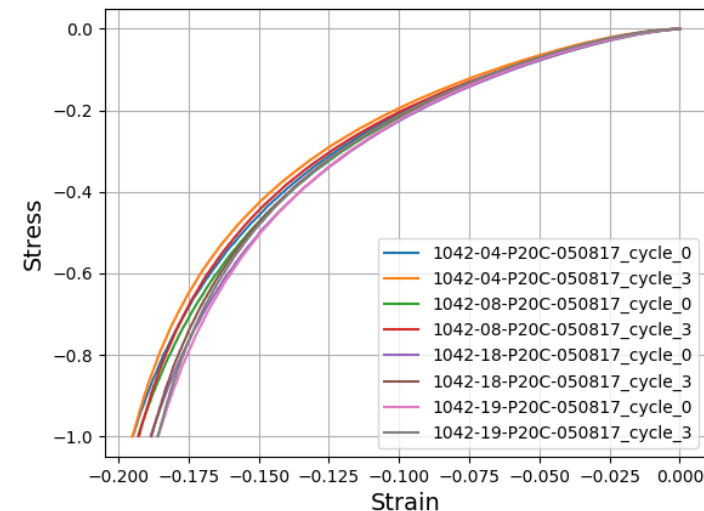
```
# Define search criteria:
list_of_IDs = [ '1042-04-P20C-050817_granta_cycles', '1042-08-P20C-050817_granta_cycles', \
                '1042-18-P20C-050817_granta_cycles', '1042-19-P20C-050817_granta_cycles' ]

# Cycles definition:
# -----
cycles_to_use = [0, 3]
#
# Functional Data:
# -----
attributes_list = [ 'Eng Stress vs. Eng Strain', \
                    'Strain [ ]', \
                    'Y min (Eng Stress vs. Eng Strain [MPa])', \
                    'Density' ]
```

Import data to Python, arrange the data

```
# Find records that meet search criteria and save them in list:
# -----
records_found = []
for _ in list_of_IDs:
    records_aux = my_tbl.search_for_records_by_text(_) [0]
    records_found.append(records_aux)

# Define dictionary with strain and stress data for each record:
# -----
data_dict = create_dictionary_from_records(records_found, attributes_list, cycles_to_use)
#
attrib = 'strain'
strain_test_Dict = create_flat_dictionary_from_nested_dict(attrib, data_dict)
#
attrib = 'stress'
stress_test_Dict = create_flat_dictionary_from_nested_dict(attrib, data_dict)
#
```



UNCLASSIFIED

# Example 2: Model Calibration

## CHIP-Foam

Constitutive equation



$$\sigma = \frac{1}{J} \frac{\partial W_{foam}}{\partial F} \cdot F^T$$

$\sigma$  – Cauchy stress

$J$  – Jacobian

$F$  – deformation gradient

$W_{foam}$  – strain energy density

Strain energy density function for CHIP-Foam formulation is given by the following formula:

$$W_{foam} = \underbrace{\frac{\hat{G}}{2}(\bar{I}_1 - 3)}_{\text{Small deformation response}} + \underbrace{\hat{K}f_1(J, J_b)}_{\text{Pore deformation (Danielsson)}} + \underbrace{C_{10}f_D(\varphi_0, \bar{I}_1, J_m) + (1 - \varphi_0)K(J_m \ln J_m - J_m + 1)}_{\text{Compressibility of parent material}}$$

Small deformation response

Pore deformation (Danielsson)

Compressibility of parent material

Material parameters such as  $\rho_s$ ,  $K$ ,  $C_{10}$ ,  $G\_hat$ , and  $K\_hat$  are determined for a foam material by fitting the model to the experimental data

UNCLASSIFIED

# Example 2: Model Calibration

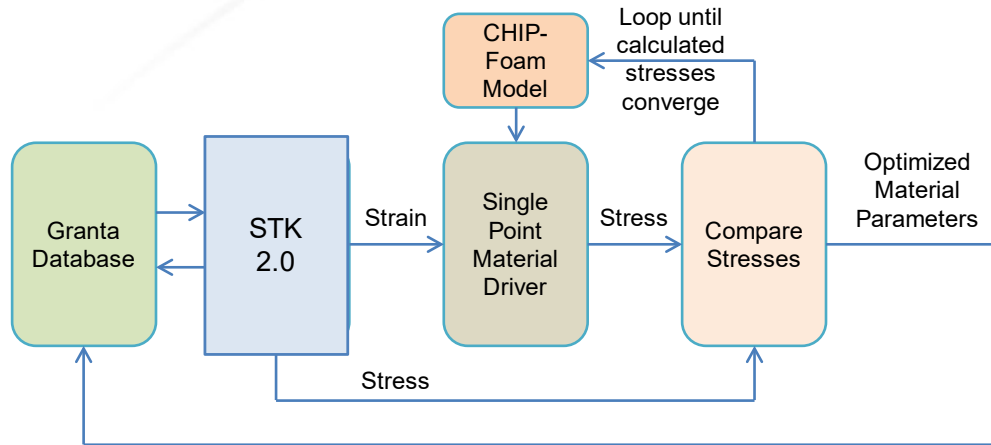
## Updates to Driver

- STK 2.0 interface added to export data from Granta and use it with the material driver
- Driver can use deformation gradient as well as strain as input
- Most commonly used deformation gradient generators implemented:
  - ✓ uniaxial strain
  - ✓ pure shear
  - ✓ simple shear
  - ✓ volumetric
- Uniaxial stress loading case added

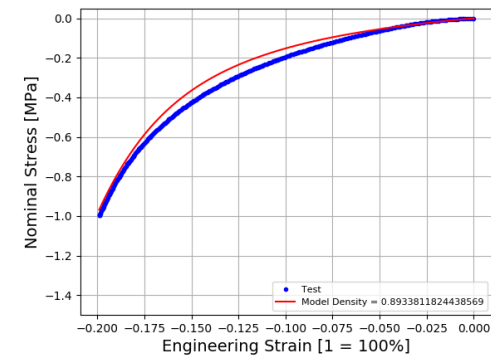
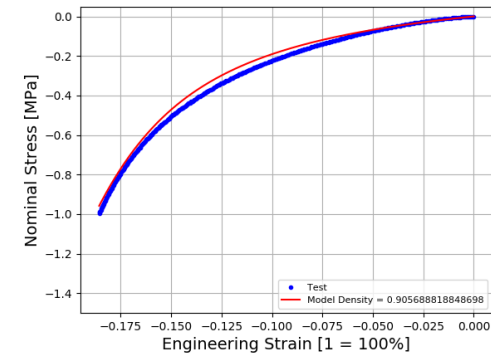
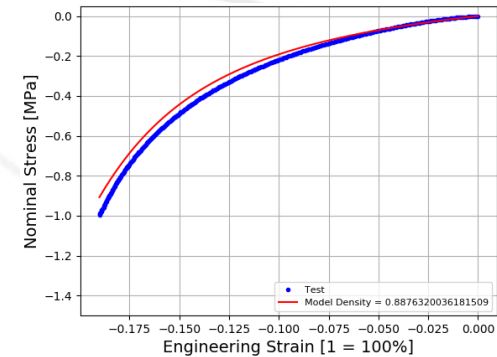
UNCLASSIFIED

# Example 2: Model Calibration

## Numerical Example



Material model calibration process is automated



UNCLASSIFIED

# Conclusions and Current Projects that Use STK 2.0

## Conclusions:

- STK 2.0 is user friendly and it is faster than GQD
- STK 2.0 is supported by Granta MI
- STK 2.0 has many additional features: plotting, changing units of individual attributes, several database searching options, and more
- Documentation of STK 2.0 exists and it is helpful
- If there is need to convert the exported data into another data structures (e.g. datanode), this can be accomplished relatively easily
- STK 2.0 can also be used to import data to database (not yet tested)

UNCLASSIFIED